Method and Apparatus For Initializing a New Node in a Network

Abstract of Disclosure

Large payload files are selectively partitioned in blocks and the blocks distributed to a plurality of distribution stations at the edge of the network qualified to have the data. Each qualified station decides how much and what portion of the content to save locally, based on information such as network location and environment, usage, popularity, and other distribution criteria defined by the content provider. Different pieces of a large payload file may be available from different nodes, however, when a user requests access to the large payload file, for example, through an application server, a virtual file control system creates an illusion that the entire file is present at the connected node. However, since only selective portions of the large payload file may actually be resident at that node's storage at the time of request, a cluster of distribution servers at the distribution station may download the non-resident portions of the file as the application server is servicing the user. The download may be in parallel and usually from the least congested nodes. New nodes added to the network learn from other nodes in the network what content they should have and download the required content, in a desired amount, onto their local storage devices from the nearest and least congested nodes without interrupting network operation. Each node manages its local storage and decides what content to prune based on information such as usage patterns.
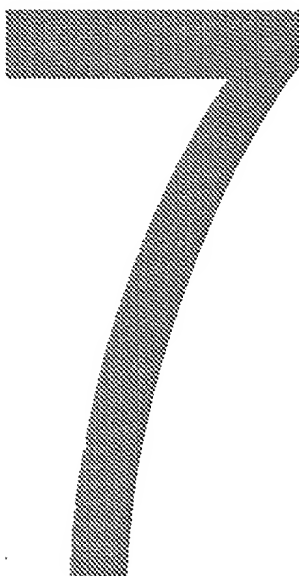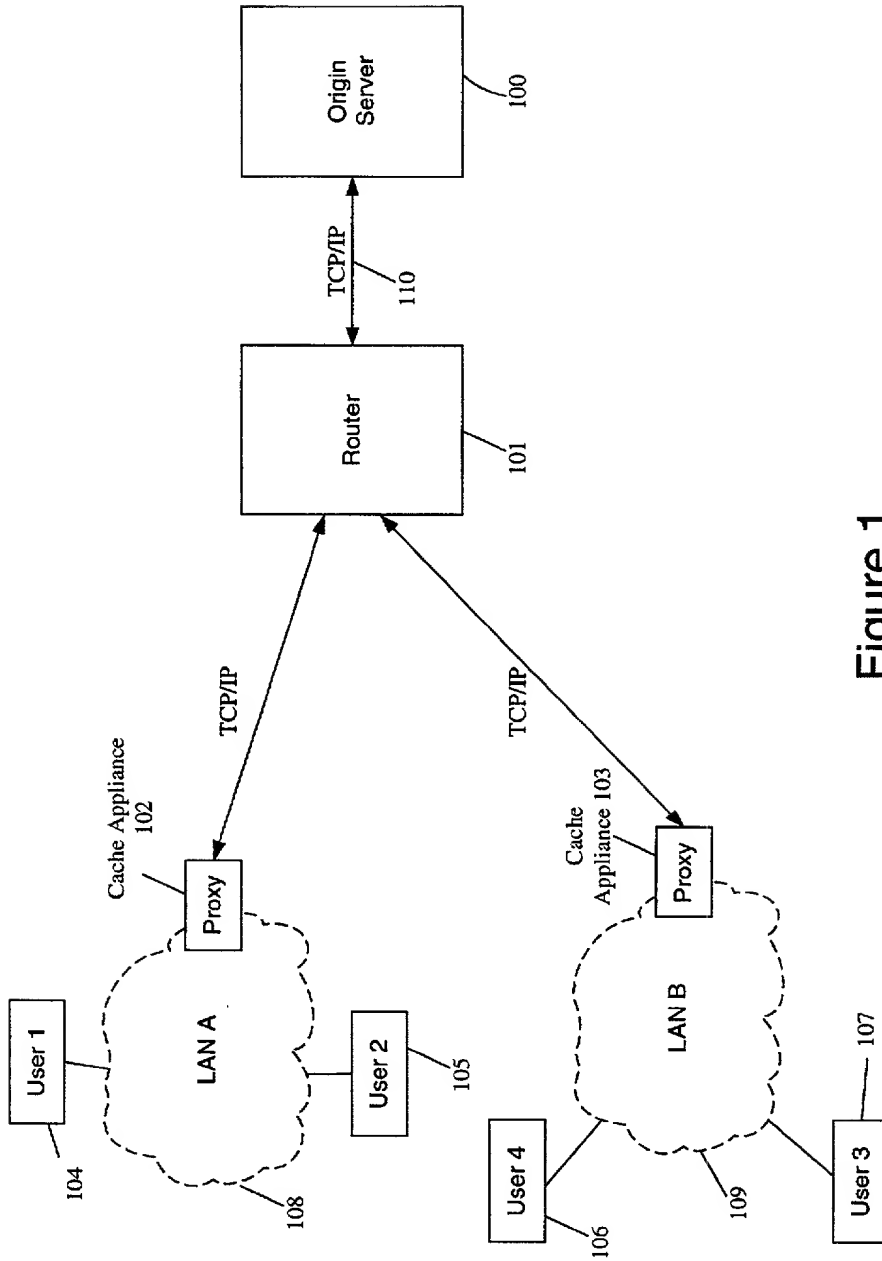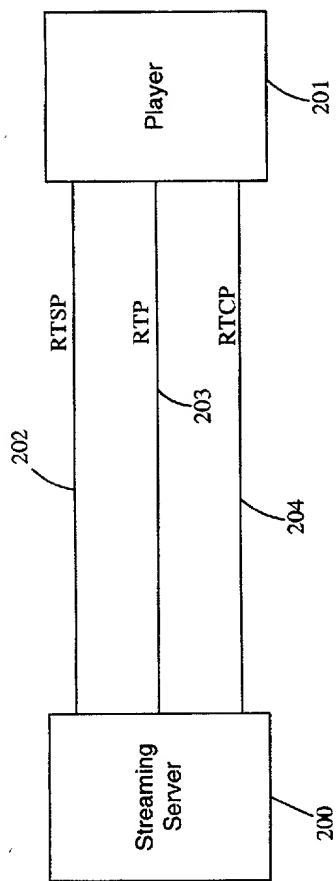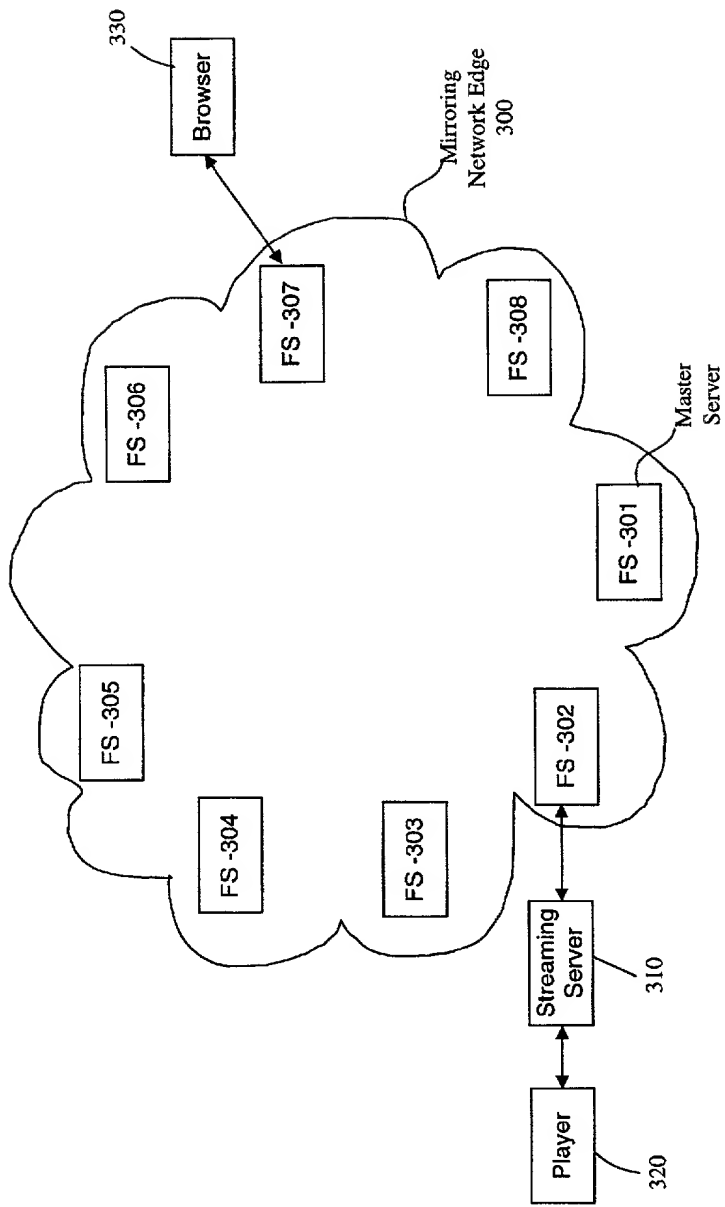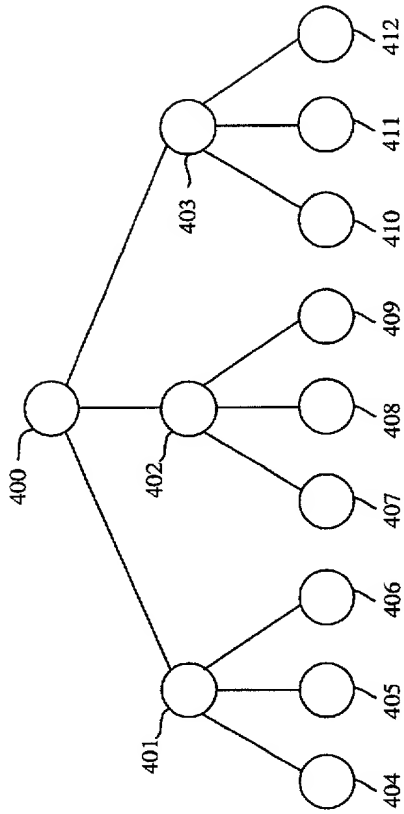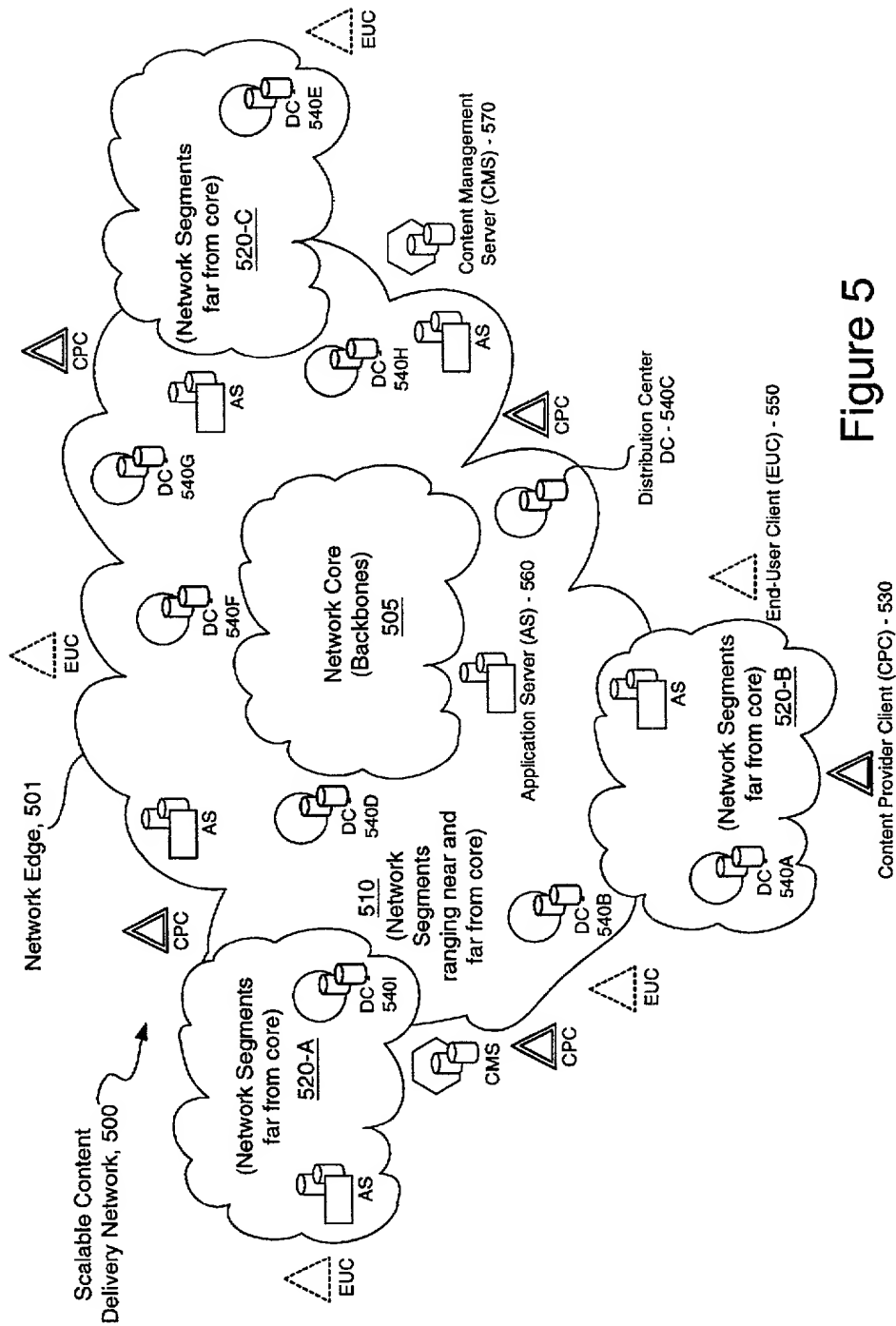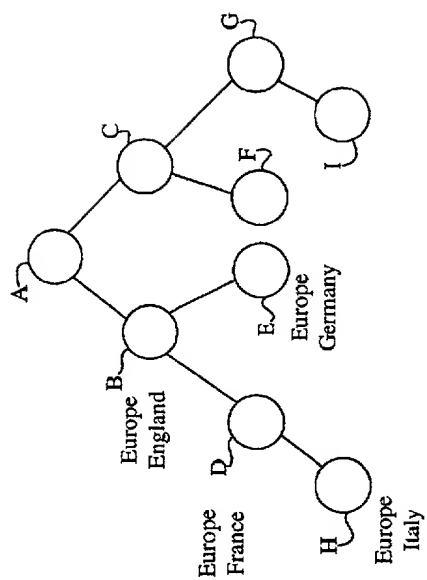
Figures

Drawings

7

Level - 2
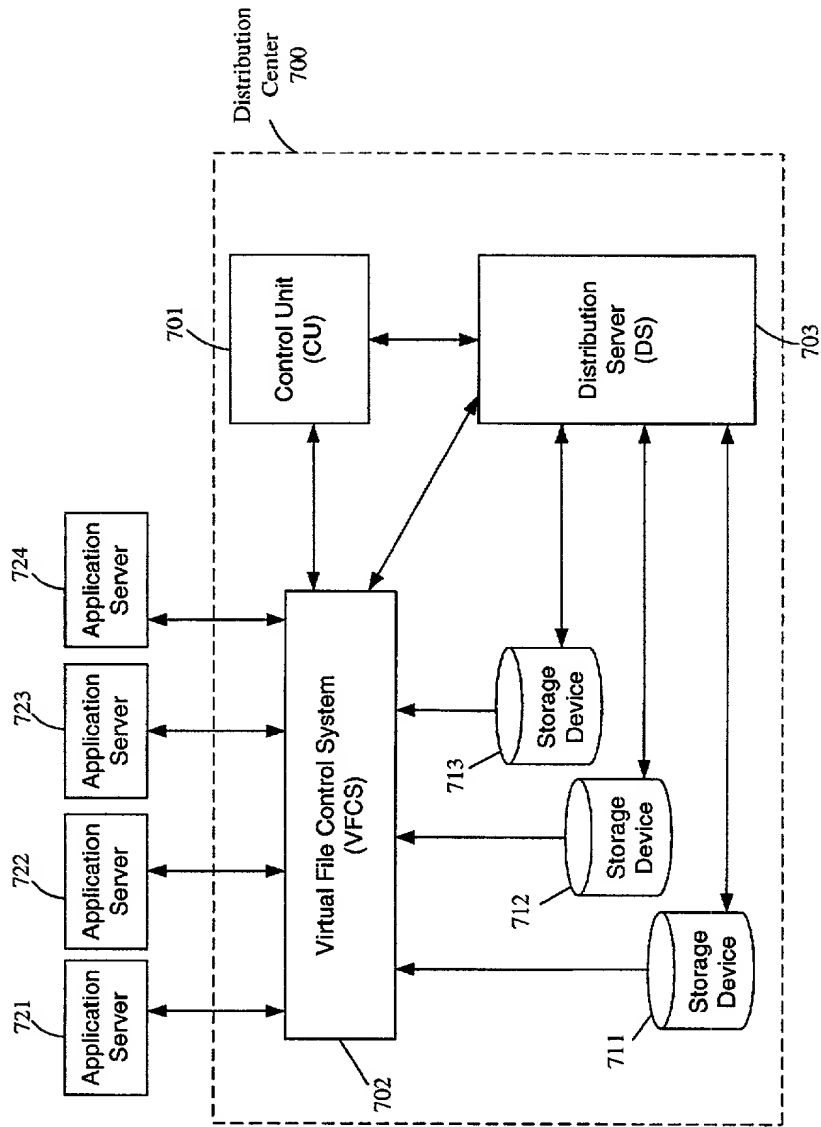Version 1.1

Figure 1

Figure 2

Figure 3

Figure 4

Figure 5

Figure 6

Figure 7
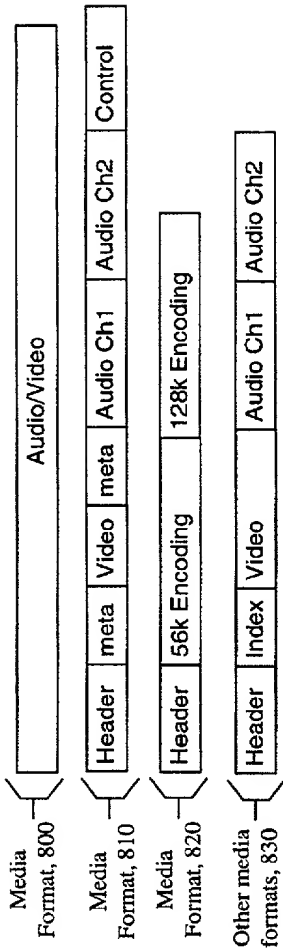
Figure 8

Figure 9

Figure 10

| Header | Video | Audio Ch1 | Audio Ch2 | Control |

Media File, 1000

| H | Video | Audio Ch1 | Audio Ch2 | C |

Track Files, 1010

H	V_1	V_2	V_3	V_4
	$A_{1,1}$	$A_{1,2}$		
	$A_{2,1}$	$A_{2,2}$		
C				
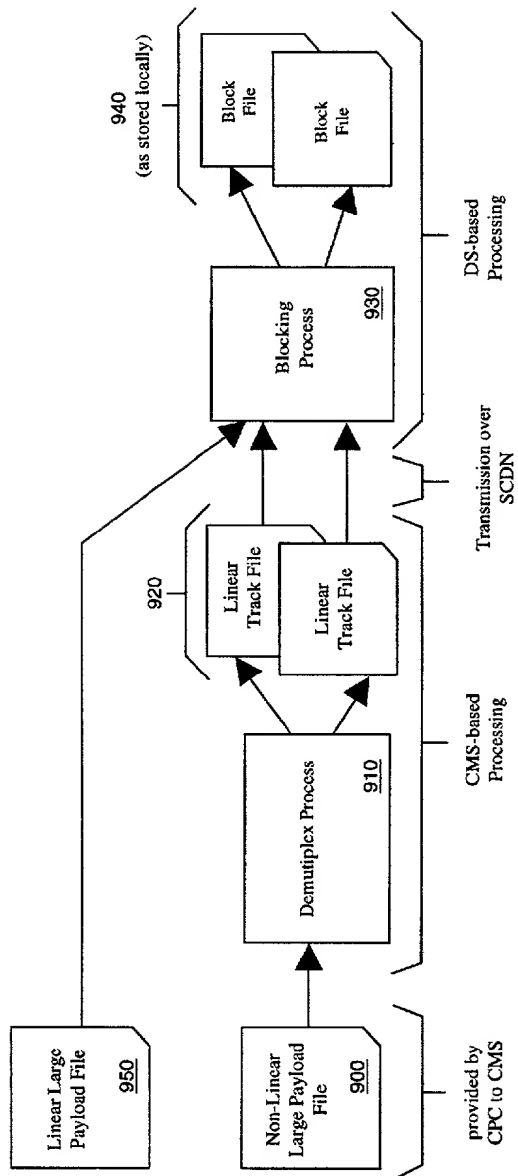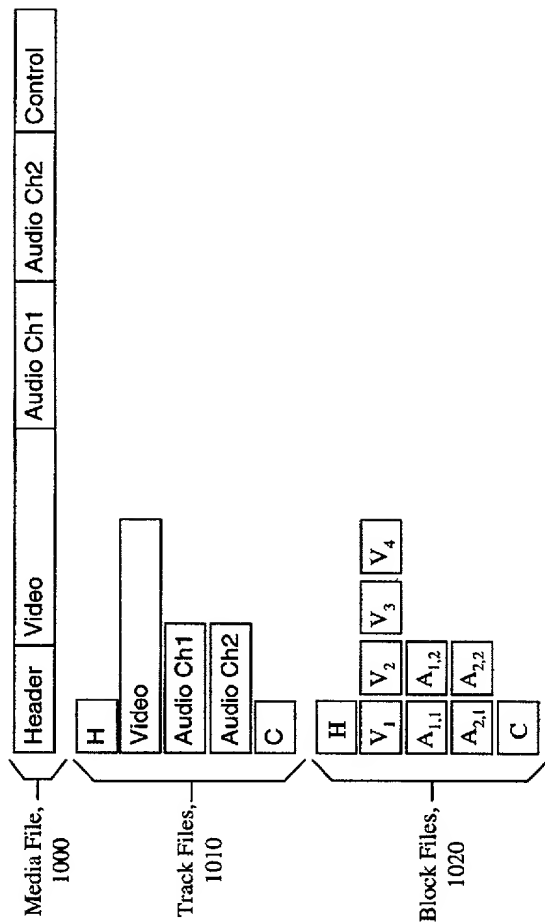
Block Files, 1020
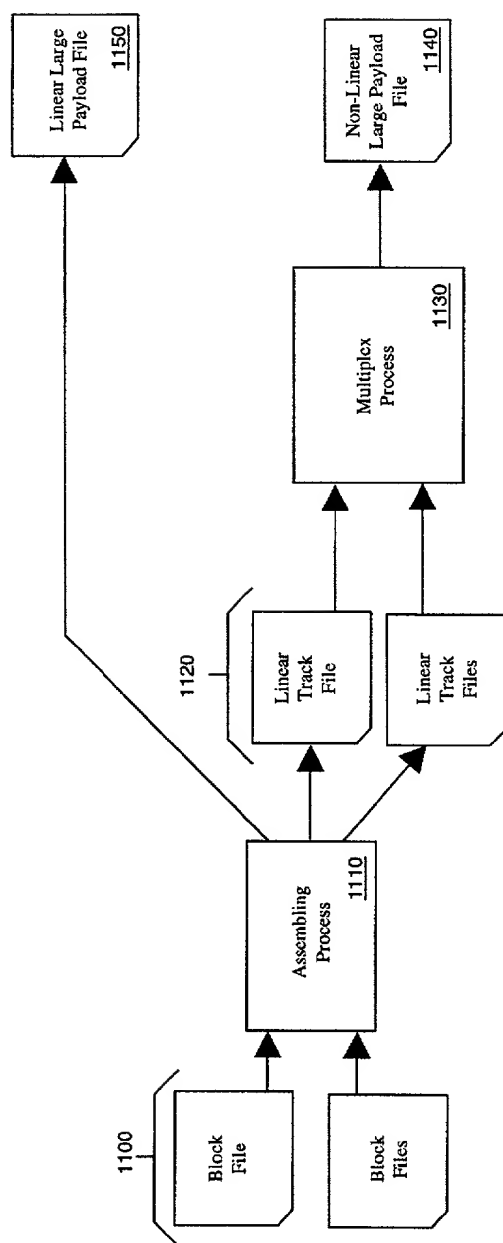
Figure 11

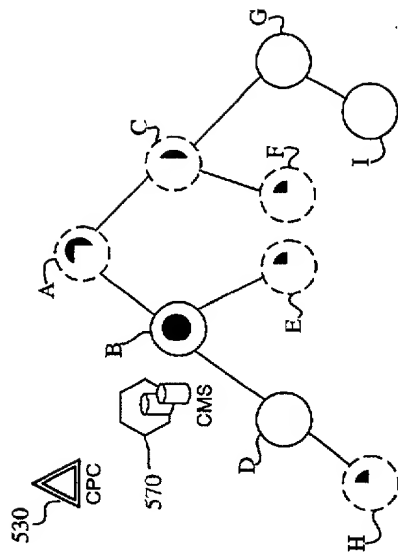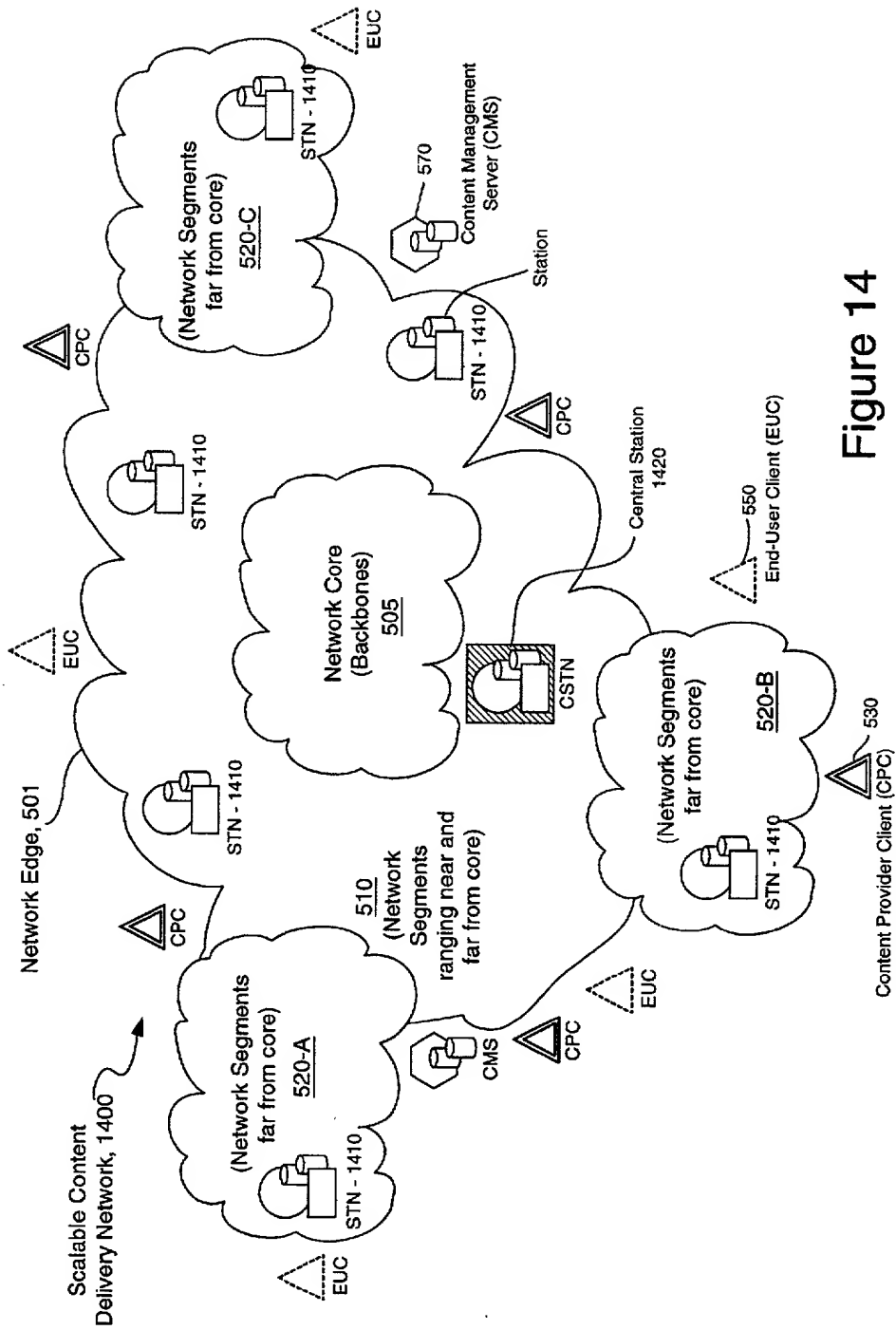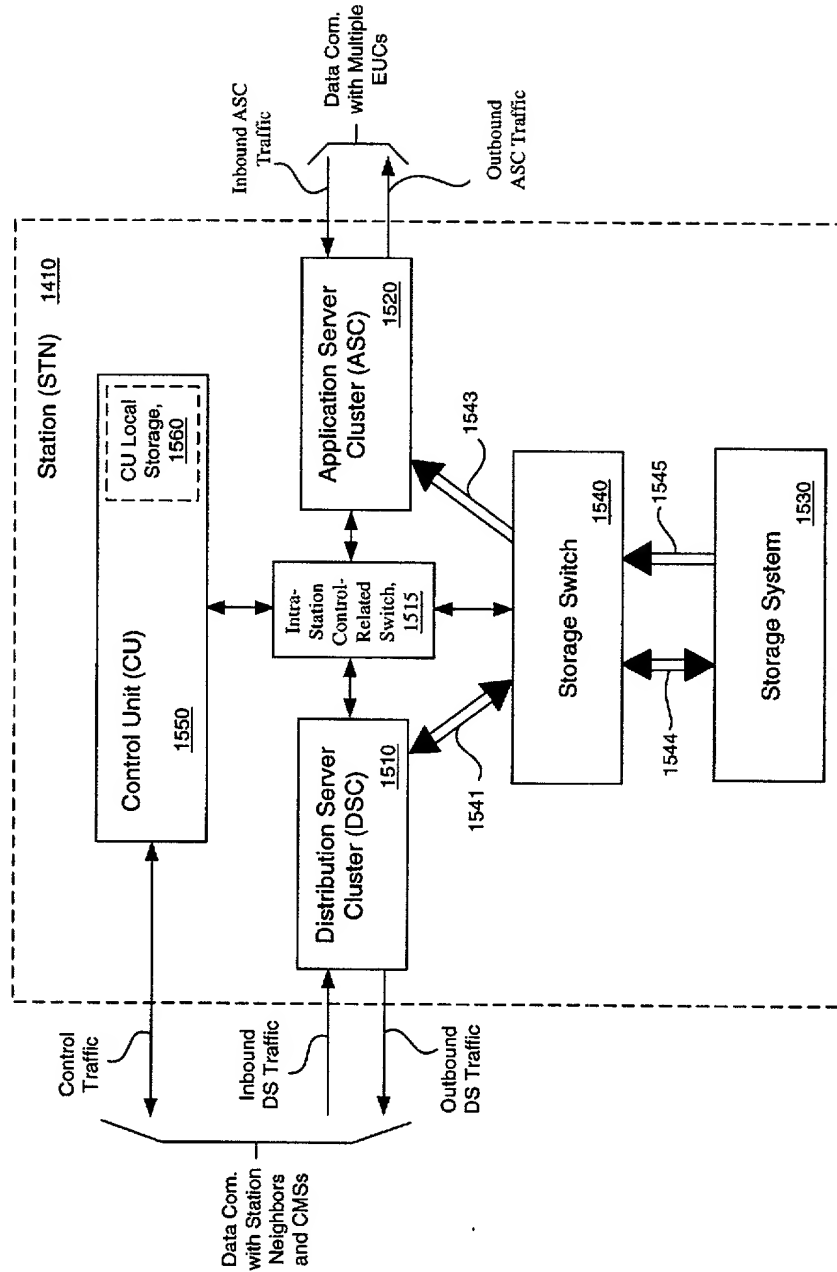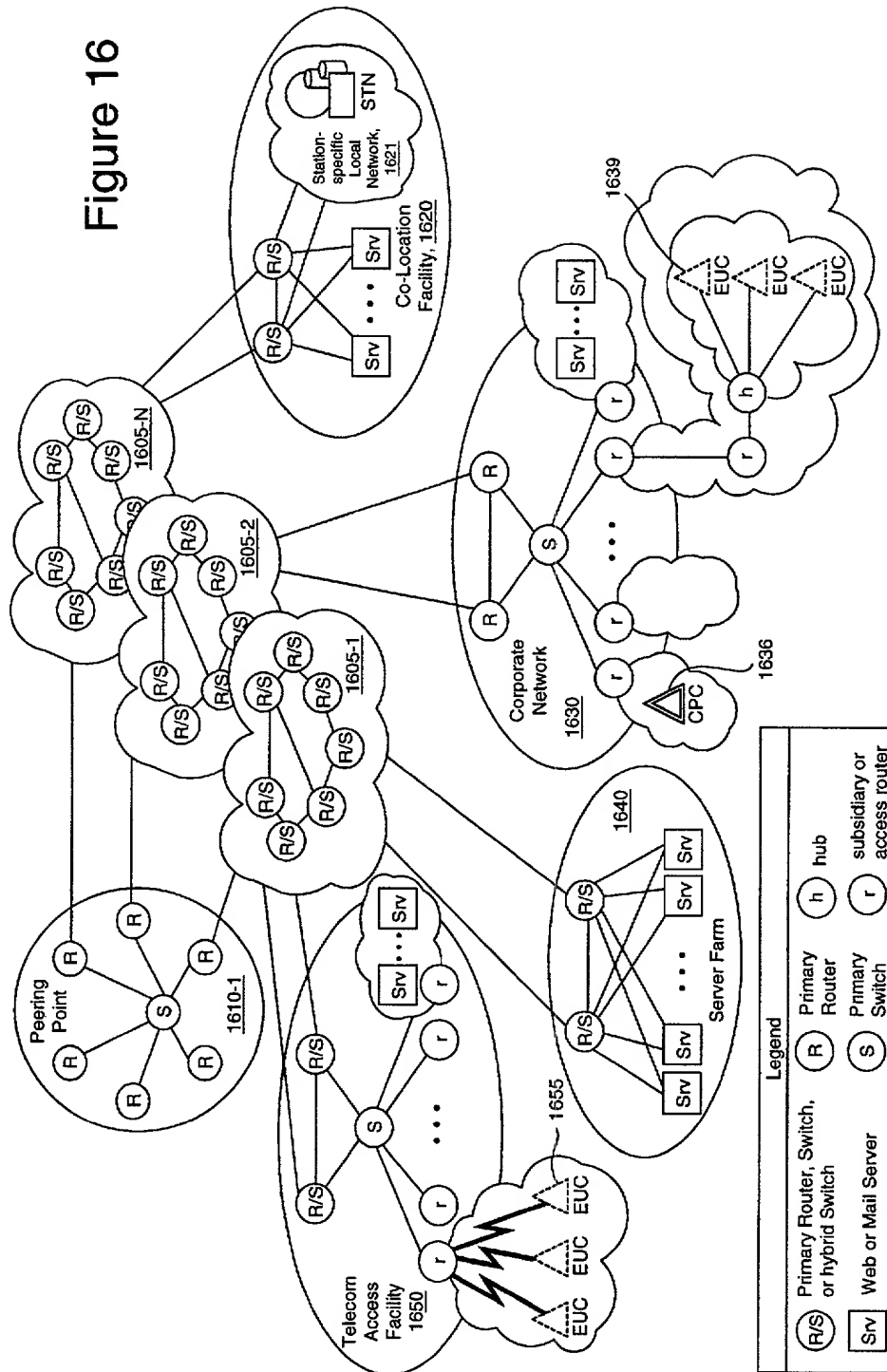| Bit Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
Attribute B	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1200
Attribute D	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1210
Attribute E	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1220
Attribute H	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1230
Rolled Up Attribute D	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1240
Rolled Up Attribute B	1	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	1250

Figure 12

Figure 13

Scalable Content Delivery Network, 1400

Network Edge, 501

EUC

CPC

STN - 1410

520-A
(Network Segments far from core)

EUC

CMS

CPC

EUC

510
(Network Segments ranging near and far from core)

Network Core (Backbones)
505

STN - 1410

EUC

CPC

STN - 1410

520-C
(Network Segments far from core)

EUC

570

Content Management Server (CMS)

Station

STN - 1410

CPC

CSTN

Central Station 1420

End-User Client (EUC)
550

520-B
(Network Segments far from core)

STN - 1410

CPC
530

Content Provider Client (CPC)

Figure 14

Figure 15

Station (STN) 1410

Control Unit (CU) 1550

CU Local Storage, 1560

Intra-Station Control-Related Switch, 1515

Application Server Cluster (ASC) 1520

Distribution Server Cluster (DSC) 1510

Storage Switch 1540

Storage System 1530

1543

1545

1541

1544

Inbound ASC Traffic

Data Com. with Multiple EUCs

Outbound ASC Traffic

Control Traffic

Inbound DS Traffic

Outbound DS Traffic

Data Com. with Station Neighbors and CMSs
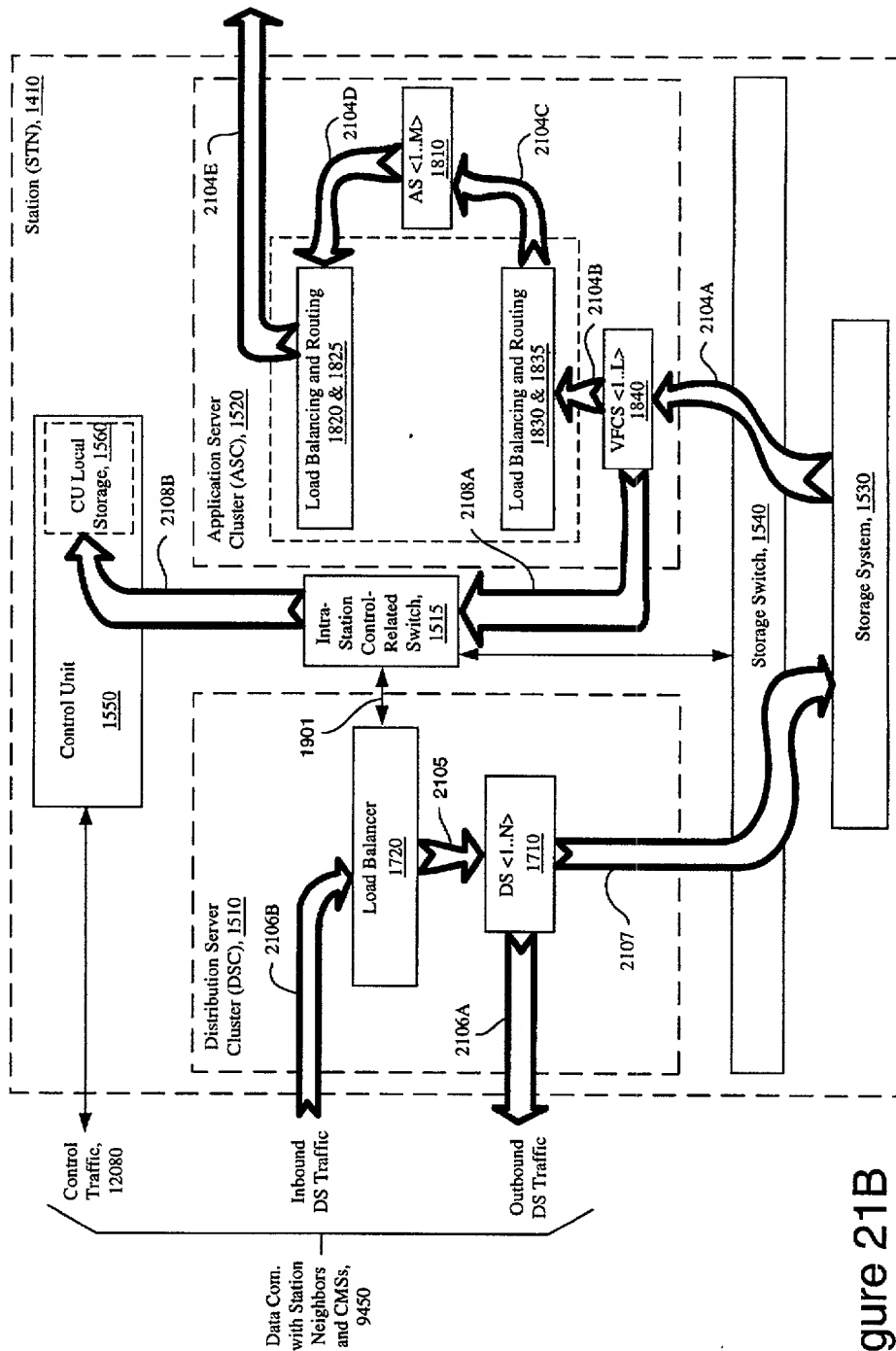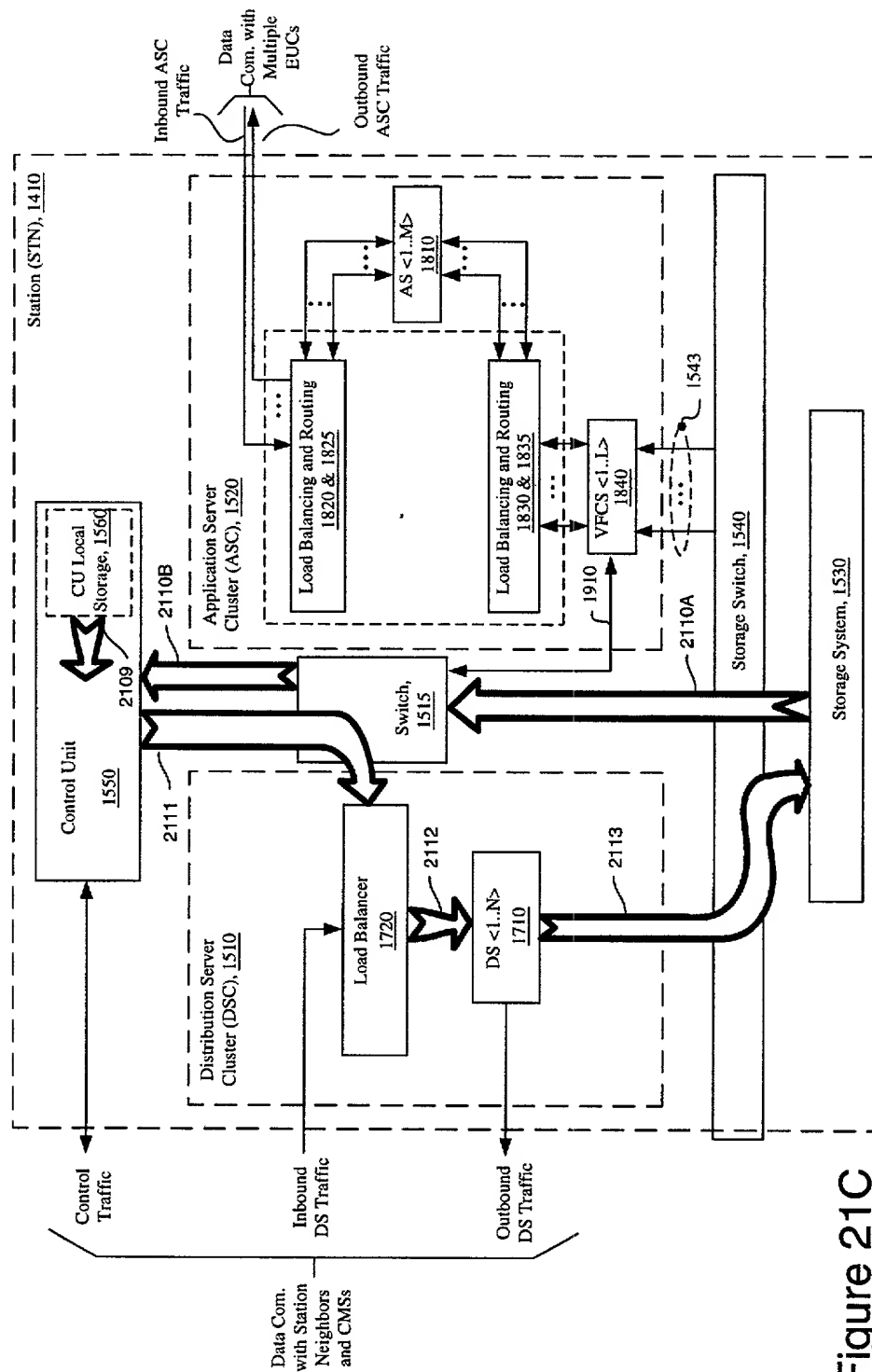
Figure 16

Figure 17

Figure 18A
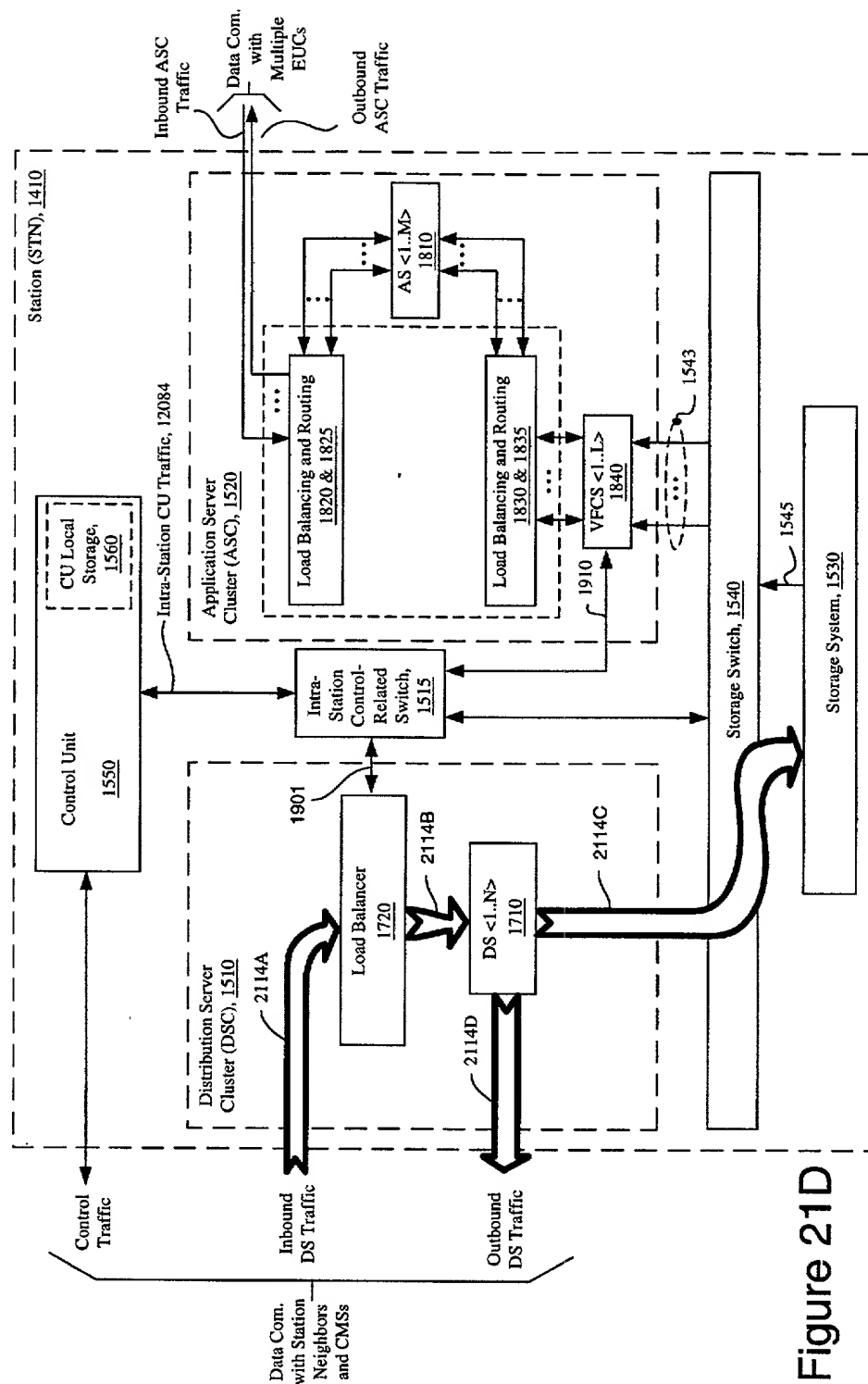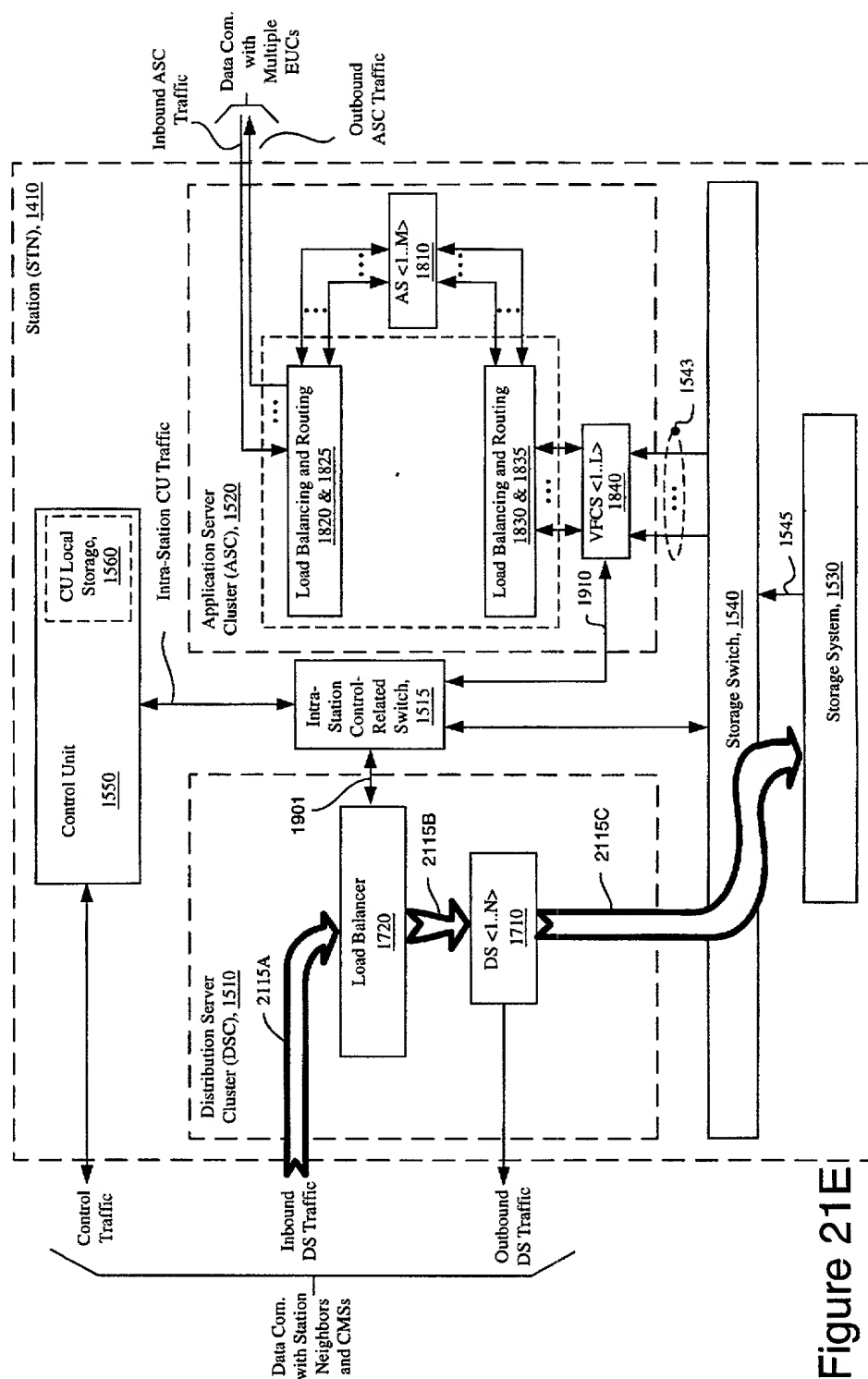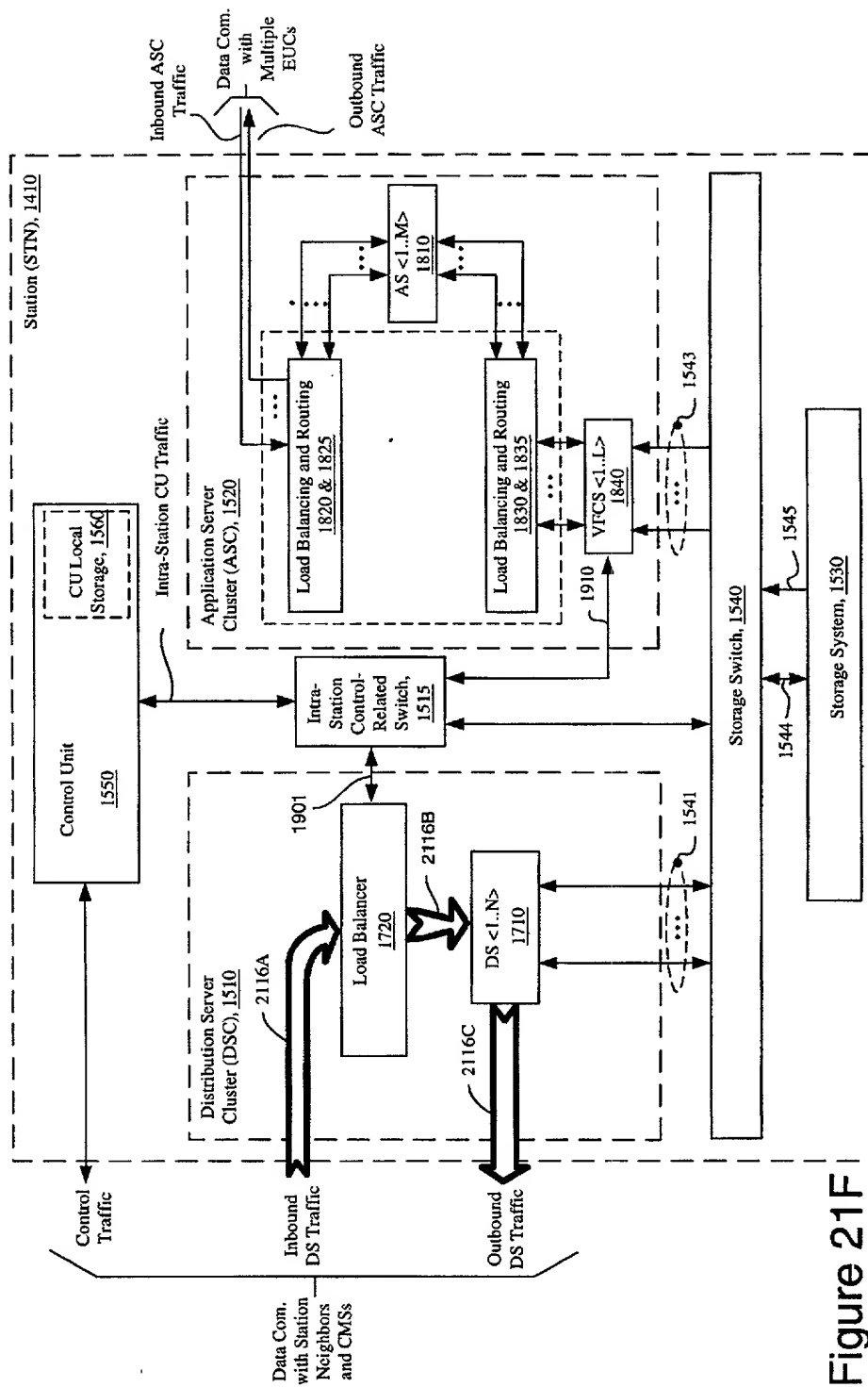
Figure 18B
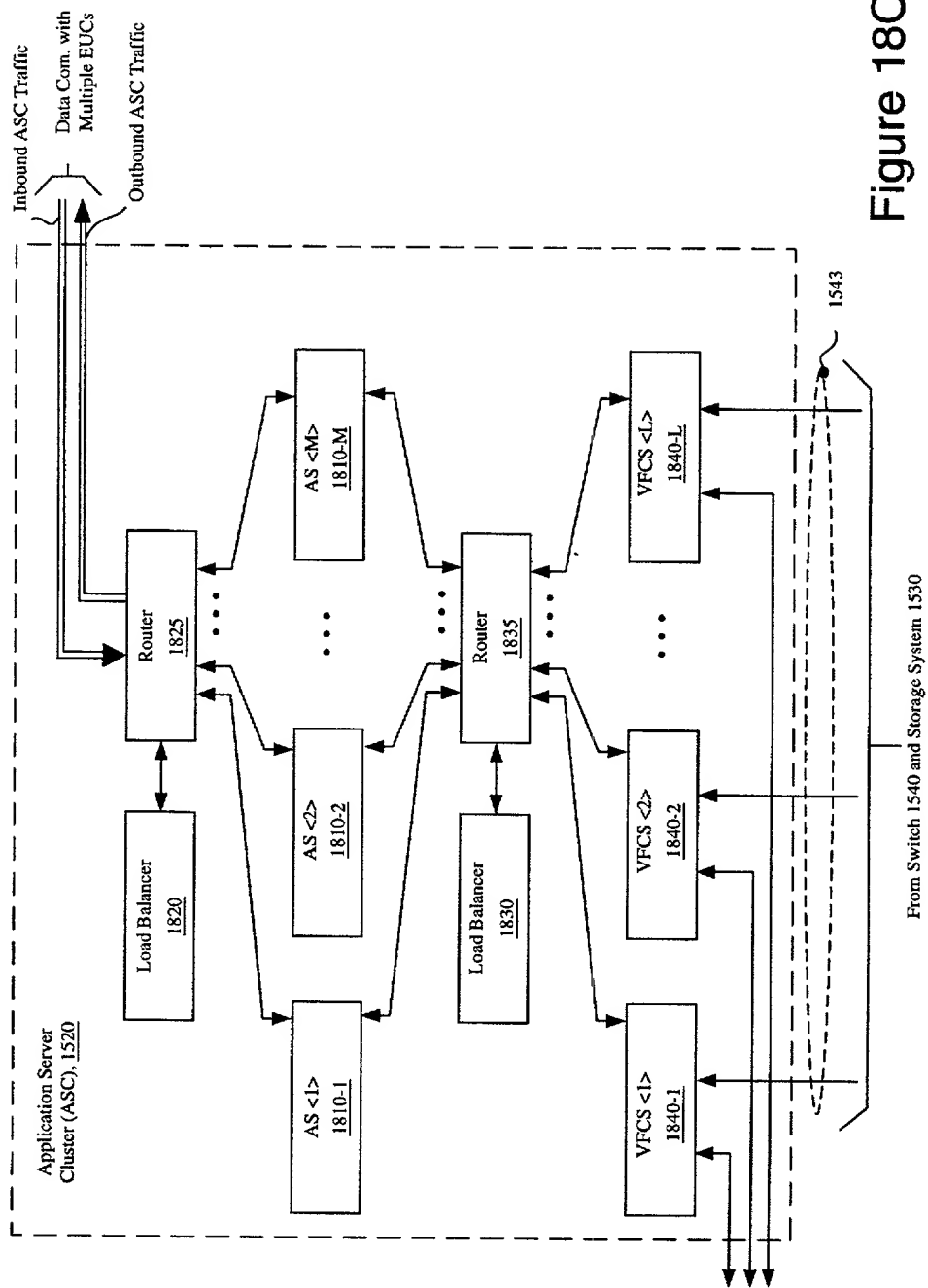
Figure 18C
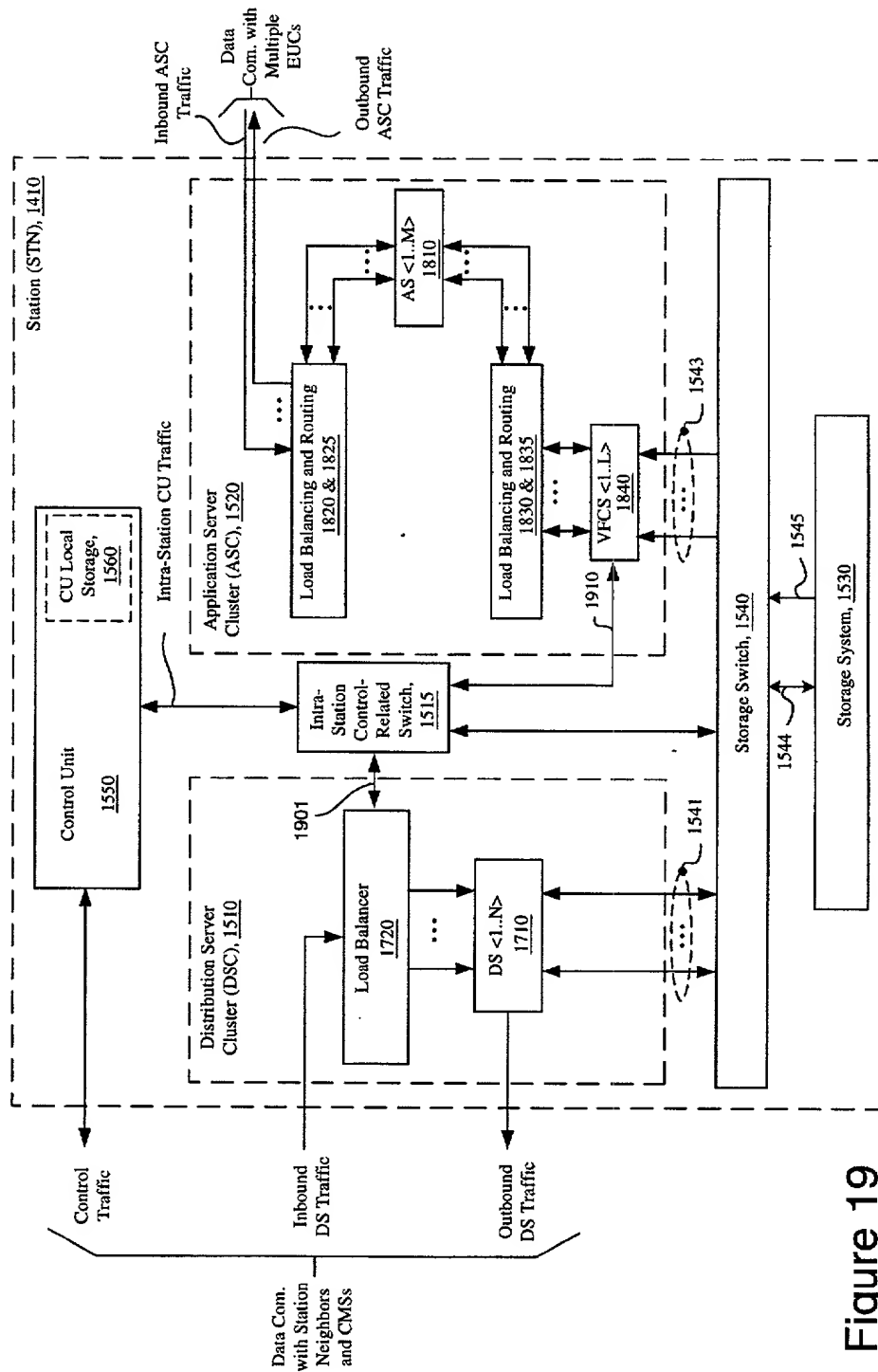
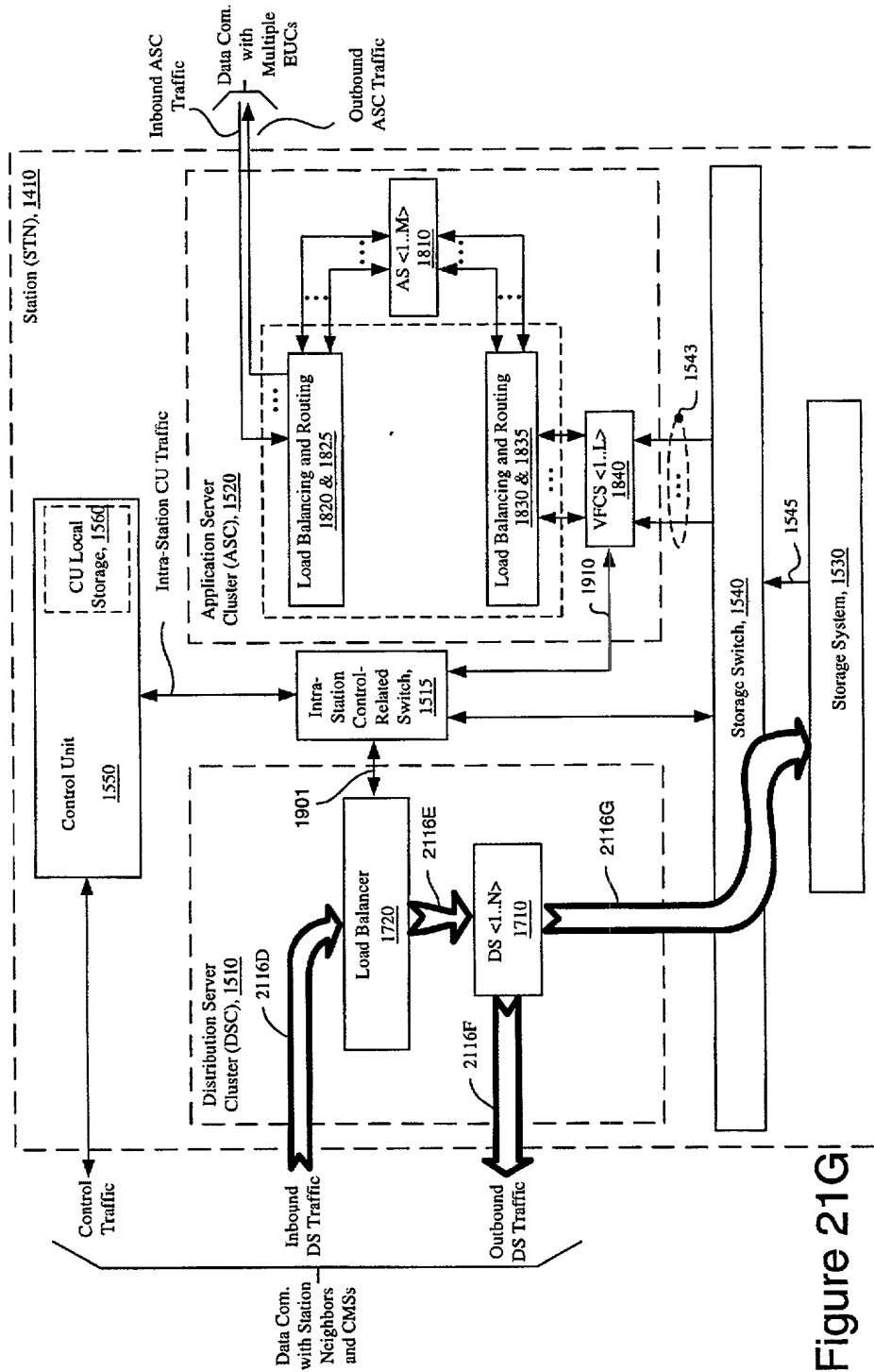Figure 19
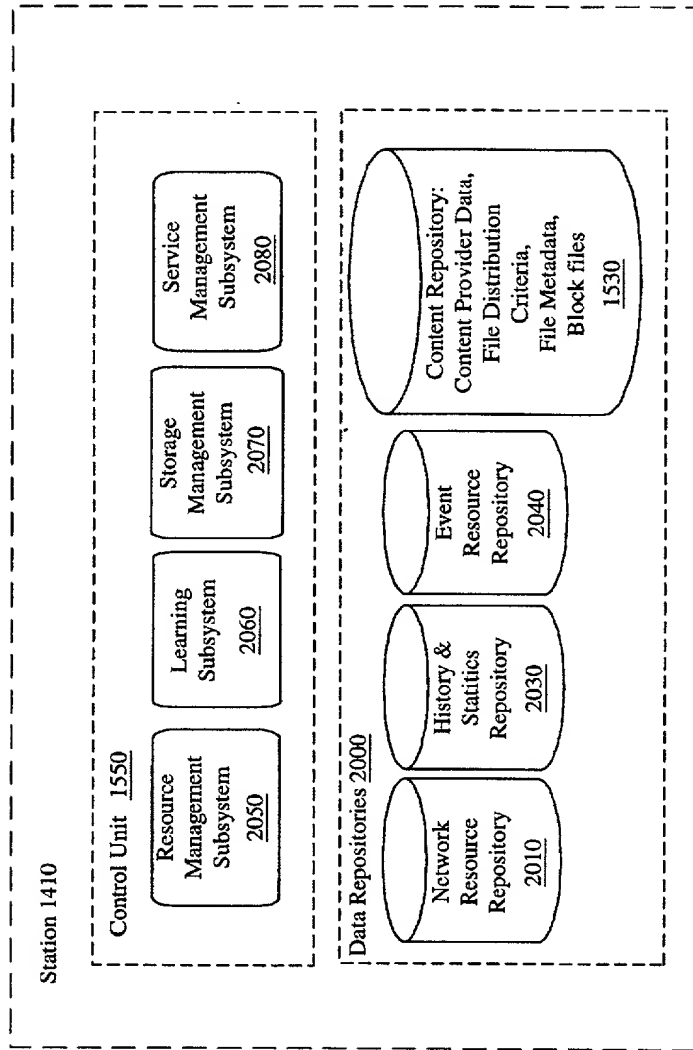
Figure 20
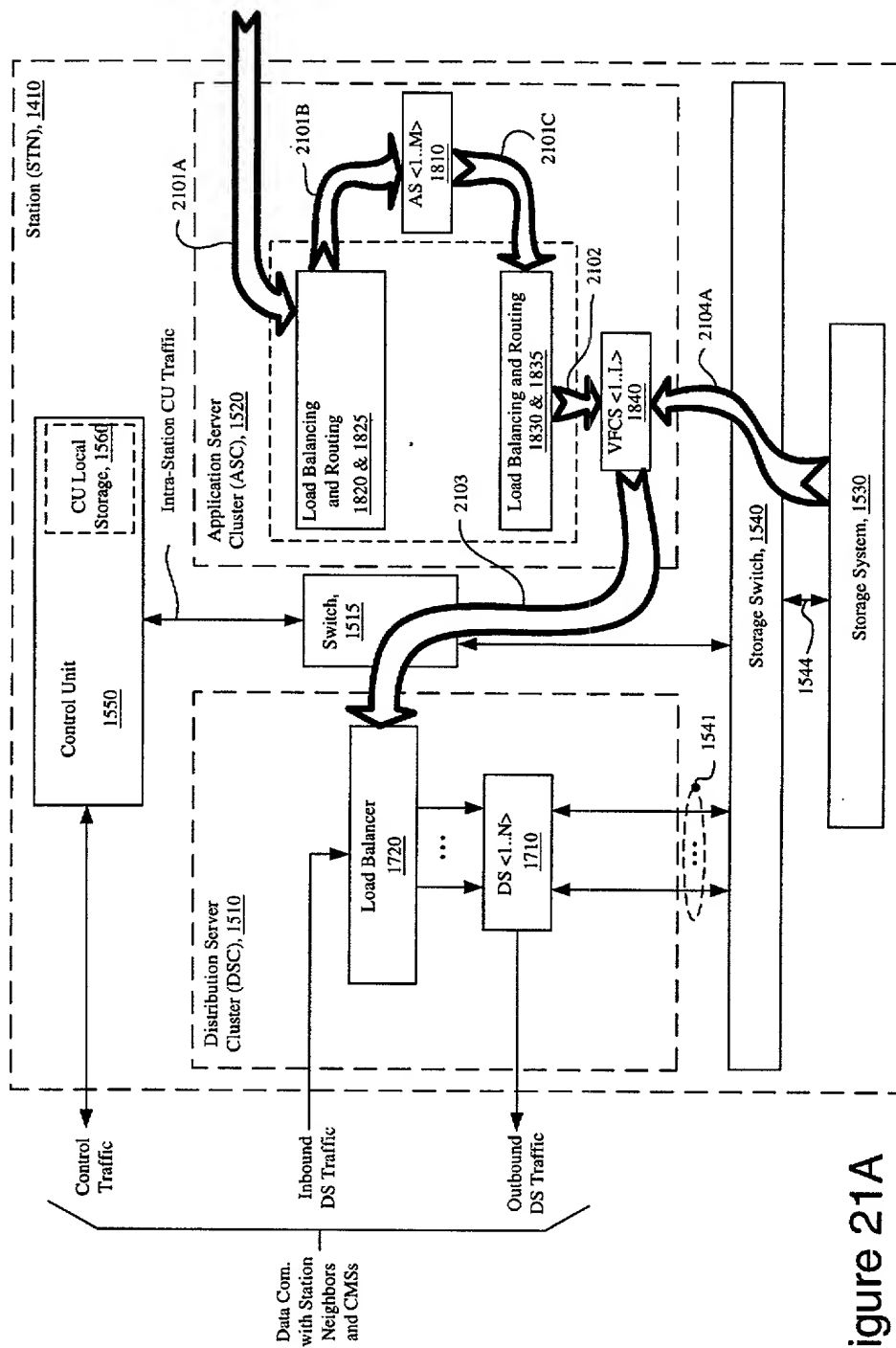
Figure 21A

Figure 21B

Figure 21C

Figure 21D

Figure 21E

Figure 21F

Figure 21G

Mount the Metadata Database and the Content Provider Data Table — 2201

Mount the Volumes that contain the File Blocks — 2202

Initialize the Content Provider Name Cache — 2203

Initialize the File System and Assign a File Handle to the Root Export Directory — 2204

Initialize the File System for the Content Provider Directories — 2205
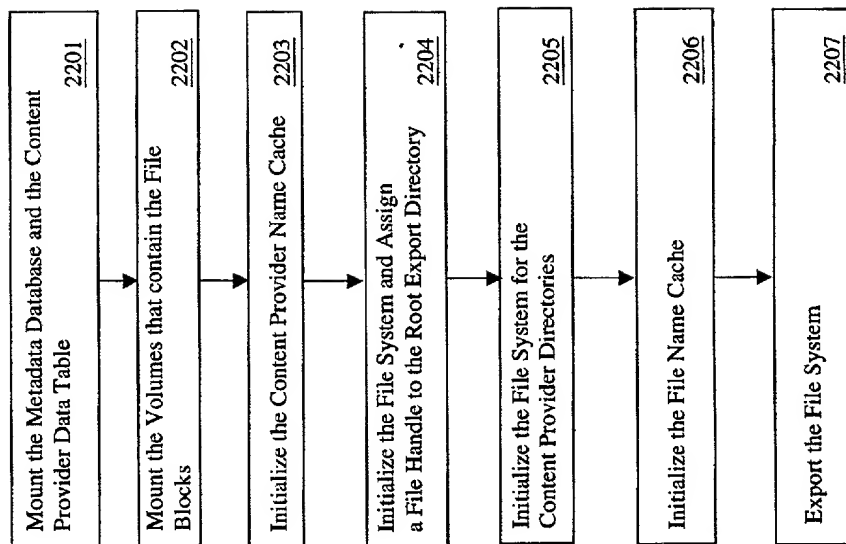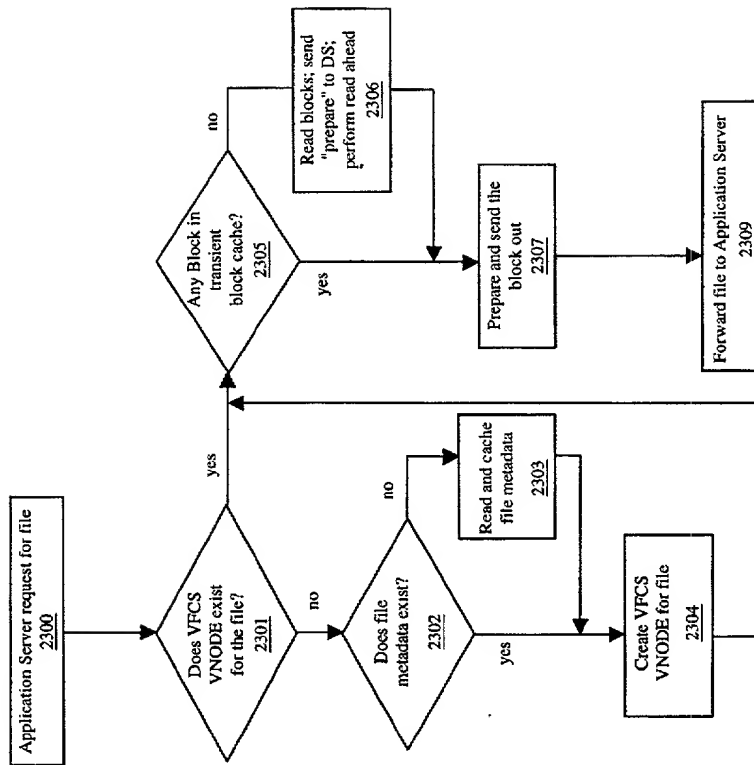
Initialize the File Name Cache — 2206

Export the File System — 2207

Figure 22

Figure 23

Figure 24

Figure 25